What is claimed is:

1. A method for scheduling a plurality of virtual machines comprising:
determining a respective resource requirement (X_i) for each virtual machine (VM);
determining a respective interrupt period (Y_i) for each VM; and
scheduling said plurality of VMs based, at least in part, on said respective X_i and Y_i values.

2. The method of claim 1 wherein, determining said respective X_i and Y_i comprises communicating said respective X_i and Y_i from an operating system (OS) running within said respective VM.

3. The method of claim 1 wherein, determining said X_i and said Y_i comprises communicating said X_i and said Y_i from an application running within an operating system (OS) running within said respective VM.

4. The method of claim 3 further comprising:
dynamically maintaining values for said X_i and said Y_i, wherein said application is a resource management application.

5. The method of claim 1 wherein, determining said X_i comprises communicating said X_i from an operating system (OS) running within said respective VM.

6. The method of claim 5 wherein, determining said X_i comprises communicating said X_i from an application running within an operating system (OS) running within said respective VM.

7. The method of claim 6, wherein said application is a resource management application, which dynamically maintains said respective X_i.

8. The method of claim 1, wherein determining a respective X_i comprises:

monitoring whether a VM reaches an idle loop;

increasing said respective X_i if said idle loop is not reached;

decreasing said respective X_i if said idle loop is reached before a predetermined percentage of said resource requirement has been utilized.

9. The method of claim 8, wherein determining respective Y_i values comprises:

filtering non-periodic interrupts;

rejecting aperiodic interrupts;

estimating said respective Y_i values for periodic interrupts; and

converging said respective Y_i values to be substantially equivalent to actual periods for said periodic interrupts.

10. An article comprising: a storage medium having stored thereon instructions that, when executed, result in a computing platform having the capability to:

schedule a plurality of virtual machines (VMs) implemented in said computing platform based, at least in part, on a respective resource requirement (X_i) and an a respective interrupt period (Y_i) for each VM of said plurality.

11. The article of claim 10, wherein said instructions, when executed result in the capability to communicate said respective X_i from an application running within a VM of said plurality.

12. The article of claim 11, wherein said instructions, when executed result in the capability to communicate said respective Y_i from an application running within a VM of said plurality.

13. The article of claim 10, wherein said instructions, when executed result in the capability to communicate said respective X_i and said respective Y_i from an operating system running within a VM of said plurality.

14. The article of claim 10, wherein said instructions, when executed result in the capability to communicate said respective X_i and said respective Y_i from a resource management application running within a VM of said plurality.

15. The article of claim 10, wherein said instructions, when executed result in the capability to determine said respective Y_i by comparing an expected interrupt period with an actual interrupt period and adjusting said respective Y_i based, at least in part, on said comparison.

16. The article of claim 10, wherein said instructions, when executed result in the capability to determine said respective X_i by detecting the occurrence of an idle loop within a VM of said plurality and adjusting X_i based, at least in part, on whether said idle loop occurs.

17. A method for determining interrupt period values comprising:
initializing said interrupt period values;
generating virtualized interrupts by virtualizing hardware interrupts;
filtering non-period interrupts;
rejecting aperiodic interrupts; and
adjusting said interrupt period values iteratively until substantially equivalent to actual interrupt periods.

18. The method of claim 17, further comprising:
acquiring resource requirement values; and
scheduling a plurality of virtual machines (VMs) to achieve real-time deadlines based, at least in part, on said interrupt period values and resource requirement values.

19. The method of claim 18, wherein said resource requirement values are acquired from said plurality of VMs.

20. The method of claim 17, further comprising determining resource requirement values, wherein determining said resource requirement values comprises:

initializing said resource requirement values; and

adjusting said resource requirement values iteratively based, at least in part, on a determination of an occurrence of a respective predetermined instruction.

21. The method of claim 20, wherein adjusting said resource requirement values comprises:

increasing said resource requirement values if said respective predetermined instruction does not occur;

decreasing said resource requirement values if said respective predetermined instruction occurs prior to a target time; and

scheduling a plurality of virtual machines (VMs) based, at least in part, on said interrupt period values and said resource requirement values.

22. An article comprising: a storage medium having stored thereon instructions that, when executed, result in a computing system having the capability to:

initialize interrupt period values;

generate virtualized interrupts by virtualizing hardware interrupts;

filter non-period interrupts;

reject aperiodic interrupts; and

adjust said interrupt period values iteratively until substantially equivalent to actual interrupt periods.

23. The article of claim 22, wherein said instructions, when executed, further result in the capability to:

acquire resource requirement values; and

schedule a plurality of virtual machines (VMs) to achieve real-time deadlines based, at least in part, on said interrupt period values and resource requirement values.

24. The article of claim 23, wherein said resource requirement values are acquired from said plurality of VMs.

25. The article of claim 22, wherein said instructions, when executed, result in said computing platform having the further capability to:

determine resource requirement values, wherein determining said resource requirement values comprises:

initializing said resource requirement values; and

adjusting said resource requirement values iteratively based, at least in part, on a determination of an occurrence of a respective predetermined instruction.

26. The article of claim 25, wherein adjusting said resource requirement values comprises:

increasing said resource requirement values if said respective predetermined instruction does not occur;

decreasing said resource requirement values if said respective predetermined instruction occurs prior to a target time; and

scheduling a plurality of virtual machines (VMs) based, at least in part, on said interrupt period values and said resource requirement values.

27. A system comprising:

a computing platform;

said computing platform being adapted to implement, at least, a virtual machine monitor (VMM) and a plurality of virtual machines (VMs);

said VMM being capable of scheduling said VMs to execute real-time applications based, at least in part, on a resource requirement (X_i) for each VM and an interrupt period (Y_i) for each VM.

28. The system of claim 27, further comprising:

an interface capable of communicating respective X_i and Y_i values for said each VM to said VMM.

29. The system of claim 27, wherein said VMM comprises:

a feedback loop capable of determining a respective X_i for said each VM;

a hardware interrupt virtualizer capable communicating device interrupts to said plurality of VMs and filtering non-periodic interrupts;

an interrupt period detector (IPD) capable of determining said periods for periodic interrupts and communicating said periods to a scheduler; and

said scheduler being capable of said scheduling of said plurality of VMs.

30. The system of claim 29, wherein said feedback loop comprises:

a detector capable of determining whether each of said VMs issues a predetermined instruction and indicating said determinations to a proportional integral derivative (PID) controller;

said PID being capable of adjusting said respective X_i for said each VM based, at least in part, on said determination and communicating said adjusted respective X_i to said scheduler.